EECS 739: Homework 4

Due: Tuesday, April 21, 2015 (At the beginning of lecture)

Answer all of the questions listed below. I will choose one problem to grade; it will be worth 20 points. The submission instructions are identical to those for Homework #1 in regards to how to submit code questions vs. non-code questions.

You may work with at most one other student in the course to formulate ideas only; please write the name of the student with whom you worked (if any) at the top of your homework solutions.

Questions:

1. Derive the Crank-Nicholson method for the following parabolic partial differential equation problem:
$$\frac{\partial u}{\partial t} = \frac{4}{\pi^2}\frac{\partial^2 u}{\partial x^2} = 0, 0 < x < 4, 0 < t;$$
$$\begin{aligned} u(0,t) &= u(4,t) = 0, 0 < t \\ u(x,0) &= \sin\left[\tfrac{\pi}{4}\left(1 + 2\cos\left(\tfrac{\pi}{4}x\right)\right)\right], 0 \le x \le 4. \end{aligned}$$
Use $h = 0.2, k = 0.04$. Write your answer as a matrix system. **Show all of your work.**

2. Consider the Leapfrog method for numerical solution of the heat equation given by
$$\frac{w_{j,n+1} - w_{j,n-1}}{2k} = \frac{w_{j+1,n} - 2w_{j,n} + w_{j-1,n}}{h^2}.$$
Determine the order of accuracy of the method.

3. Develop and implement in C++ a serial Crank-Nicholson scheme for the system of two coupled PDE's $u_t = au_{xx} - cu + dv, v_t = bv_{xx} + cu - dv$. These equations specify a "reaction-diffusion" system. Here, u and v are chemical concentrations distributed on a linear segment. Chemical u diffuses, and it also changes to v with rate c. Chemical v also diffuses and changes to u with rate d. Use $[0,1]$ for the spatial domain, and integrate to $t = 1/2$. Use as initial conditions
$$u(x,0) = \begin{cases} 1, & x \le 0.5 \\ 0, & x > 0.5 \end{cases}$$
and $v(x,0)$ is the complement, i.e., $v(x,0) = 1 - u(x,0)$. Take $a = 1, b = .1, c = .5, d = .2$. Note that for the terms $-cu + dv$ and $cu - dv$, C-N is like a trapezoid method for ODE's. (Please look up the trapezoid method if you are not familiar with it.)

   Note that to take a time step with C-N, you must solve a system of linear equations. (To determine the coefficient matrix and right-hand side of the linear equations, rewrite C-N with all the step-$n+1$ terms on the left-hand side and all the step-n terms on the right. Then you will see the linear equations that must be solved.) Write the linear system you obtain in matrix form. Solve the linear system of equations in serial using the Thomas algorithm. (Note that you will be able to make the calculation efficient by making use of the structure of A.)

   Use as boundary conditions $u_x(0,t) = v_x(0,t) = u_x(1,t) = v_x(1,t) = 0$. These boundary conditions are implemented by assuming that u and v evaluated at $x-$ coordinates one step past the boundary on ether side (which enter into the C-N formula for the nodes on the boundary) are equal to the adjacent u and v at the boundary.

Notice there are some nonphysical spikes in the computed solution at $t = 1/2$. Experiment with different values of h and k to see what happens to the non-physical spikes.

Hand-in at least two interesting plots and a brief description of what you determined experimentally. **This is in addition to the items which you normally turn in for serial implementation questions.**

4. Develop and implement in MPI and C++ the parallel red-black Gauss-Seidel algorithm as described in Section 7.2.4 (starting on p. 383) in the textbook by Karniadakis and Kirby.

   Apply your algorithm to numerically solve the following partial differential equation in parallel:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \left(x^2 + y^2\right) e^{xy}, 0 < x < 2, 0 < y < 1;$$

$$u(0, y) = 1, u(2, y) = e^{2y}, \quad 0 \leq y \leq 1;$$

$$u(x, 0) = 1, u(x, 1) = e^{x}, \quad 0 \leq x \leq 2.$$

   Experiment with solving the above PDE with $m, n = 10, 50, 100$ mesh points. Demonstrate that your algorithm obtains correct results by printing the value of the norm of the residual vector upon convergence for each different mesh.

   Test your algorithm for strong scaling on up to 16 nodes.

   Turn-in a bar plot illustrating your strong scaling results. **This is in addition to the items which you normally turn in for parallel implementation questions.**